bloopark

bloopark wants you > Working Student (m/w/d)

Dear Applicant.

We are very pleased to invite you to the shortlist for our Working Student position. We want to make sure that we hire the right person and also you as an applicant will definitely want to get to know us better and for that we have created this test case.

WORKING STUDENT TEST CASE

Deadline (when to deliver): within 7 days, after receiving the test case mail
Send Mail with your solution to: nasser@bloopark.de , anne@bloopark.de , moustafa@bloopark.de
Reference: "Working Student Test Case 2022"

How to deliver:

Create a GitHub repo of your solution and share the repository with us as your final solution.

We tried to give you as much information as possible, but if there is something missing you can add it from your own imagination. Here at Bloopark we really like that :)
Also, it would be better if you can complete the full test case, but do not feel bad if there is something you can not deliver. Just enjoy this test and share with us your results!

Test Case:

Python

A university's Office of Admission keeps track of student majors. Each student can have a single major. Below is an example of how their system stores students, majors, and how it manipulates them:

```
students = [("Allen Anderson", "Computer Science"),
            ("Edgar Einstein", "Engineering"),
            ("Farrah Finn", "Fine Arts")]

def add_new_student(students, name, major):
    students.append((name, major))

def update_student(students, index, name, major):
    students[index] = (name, major)

def find_students_by_name(students, name):
    return [student for student in students if name in student[0]]

def get_all_majors(students):
    return [student[1] for student in students]
```

What can be concluded from the code snippet above? *(Select all acceptable answers.)*

1. In the *update_student* function, the '(' and ')' parentheses can be removed without causing any errors.
2. Calling *find_students_by_name(students, 'in')* returns a list of 2 tuples.
3. The *add_new_student* function can be rewritten as seen below and still maintain identical functionality: students[len(students)] = (name, major)
4. Calling *get_all_majors(students)* returns a list of 3 tuples.
5. The *add_new_student* function adds a new student in the last place in the list.
6. The name of the first student in the array can be set to the *new_name* variable, like *students[0][0] = new_name*

bloopark

Python and SQLite

Given the following data definitions: *(Use Python and SQLite only)*

1. Create Python classes that represent these data definitions
2. Write a method that returns the number of students whose first name is John.
3. Assuming that the table containing the students enrolled in a yearly course has incorrect data in records with ids between *20* and *100*. Write method that updates the field 'year' of every faulty record to 2015.

```
TABLE students
   id INTEGER PRIMARY KEY,
   firstName VARCHAR(30) NOT NULL,
   lastName VARCHAR(30) NOT NULL

TABLE enrollments
  id INTEGER NOT NULL PRIMARY KEY
  year INTEGER NOT NULL
  studentId INTEGER NOT NULL
```

XML

1. Extract data from this XML document with python (xml library)
2. Add the price in the document with python

```
<?xml version="1.0" encoding="UTF-8"?>
<metadata>
<food>
    <item name="breakfast">Idly</item>
    <description>Two idly's with chutney</description>
    <calories>553</calories>
</food>
<food>
    <item name="breakfast">Paper Dosa</item>
    <description>Plain paper dosa with chutney</description>
    <calories>700</calories>
</food>
</metadata>
```

Just a little reminder:

Deadline (when to deliver): within 7 days, after receiving the test case mail
Send Mail with your solution to: nasser@bloopark.de , anne@bloopark.de , moustafa@bloopark.de
Reference: "Working Student Test Case 2022"
How to deliver:

Create a GitHub repo of your solution and share the repository with us as your final solution.

We wish you good luck and are looking forward to your answer!

All the best for you!
The bloopark Team